



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/747,091	12/21/2000	Jeffrey L. Kodosky	5150-45900	1583
35690	7590	05/20/2004	EXAMINER	
MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.			KANG, INSUN	
P.O. BOX 398			ART UNIT	
AUSTIN, TX 78767-0398			PAPER NUMBER	

2124
DATE MAILED: 05/20/2004

/0

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/747,091

Applicant(s)

KODOSKY ET AL.

Examiner

Insun Kang

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 04 March 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-31 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-31 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the amendment filed 3/4/2004.
2. As per applicant's request, claim 23 has been amended. Claims 1-31 are pending.

Double Patenting

3. The applicant fails to show the reasons to traverse the double patenting rejection. Therefore, the rejection of double patenting is maintained.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-10, 12-17, and 19-31 are rejected under 35 U.S.C. 102 (e) as being anticipated by Uczekaj et al. (US 5,920,718).

Per claim 1, Uczekaj et al. disclose:

A computer-implemented method for programmatically generating graphical program based on a state diagram comprising: receiving state diagram information ("graphical control system for automatically generating application program shell code (col 3, lines 33-49)"; "user to enter objects with state information...based on

the entered information, generates application shell code (col 4, lines 28-39)
wherein the state diagram information specifies one or more states ("a minimum of one data state is required for describing control information (col 1, lines 39-41)"
programmatically generating the graphical program in response to the state diagram information(col 4, lines 28-39)
wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the specified one or more states (col 1, lines 39-41; col 3, lines 33-49; col 4, lines 28-39; col 6, lines 1-23; col 10, lines 7-40; col 16, lines 17-23).

Per claim 2, Uczekaj et al. disclose:

The method of claim 1, wherein the state diagram information represents a state diagram (col 10, lines 7-22).

Per claim 7, Uczekaj et al. disclose:

The method of claim 1, wherein said programmatically generating the new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating (col 3, lines 34-46, "...eliminates the need to program in or edit control application code by hand...").

Per claim 8, Uczekaj et al. disclose:

The method of claim 1, wherein said programmatically generating the graphical program comprises programmatically generating a block diagram including the graphical source code corresponding to the specified one or more states.

In col 10, lines 7-22, Uczekaj et al show that "after the user has completed entry of the information ...displaying the state diagram ...in a predefined location on the display for presenting a state diagram... state names are displayed in ovals representing object states." In col 15, lines 7-14, Uczekaj et al specifically show

the block diagram "displayed as oval...the lines with arrows that interconnect the oval states indicate transitions from one state to the other in the direction of the arrows...."

Per claim 9, Uczekaj et al. disclose:

The method of claim 1, wherein the programmatically generated graphical source code includes placeholder graphical source code for each state. The placeholder (e.g. dummy, container) is for the user to fill in the program with specific instruction code. Uczekaj et al. show that "the generated code is called application shell code because all code is generated except the specific code for any user method names entered in define user method ... (col 10, lines 27-40)". See also col 6, lines 30-33.

Per claim 10, Uczekaj et al. disclose:

The method of claim 9, further comprising: for each state, a user manually entering graphical source code specifying execution instructions to be performed when the state is active during execution of the graphical program (col 6, lines 30-33; col 10, lines 27-40; col 9, lines 56-64).

Per claim 12, Uczekaj et al. disclose:

The method of claim 1, wherein, for at least one state, the state diagram information specifies program code associated with the state (col 3, lines 4-8; col 9, lines 55-65); wherein the

programmatically generated graphical source code includes the specified program code (col 4, lines 28-39; col 6, lines 30-33, col 5, lines 26-40).

Per claim 13, see the rejection of the claim 12 above.

Per claim 14, Uczekaj et al. disclose:

The method of claim 1, wherein the state diagram information further specifies one or more state transitions (col 1, lines 40-48), wherein each state transition specifies a transition from a first state to a second state (col 9, lines 45-64; col 15, lines 7-14); wherein said programmatically generating further comprises programmatically generating graphical source code corresponding to the specified state transitions (col 10, lines 23-40).

Per claim 15, see the rejection of claim 9 above.

Per claim 16, Uczekaj et al. disclose:

The method of claim 15, further comprising: for one or more state transitions, a user manually entering graphical source code specifying a Boolean condition associated with the state transition (col 1, lines 39-48; col 9, lines 46-67; col 10, lines 1-6).

Per claim 17, Uczekaj et al. disclose:

The method of claim 14, wherein the state diagram information specifies at least two state transitions from a particular state (col 15, lines 7-14); wherein the state diagram information also specifies a priority ordering for the at least two state transitions (col 15, lines 30-33); wherein said programmatically generating comprises programmatically generating graphical

source code such that, during execution of the graphical program, Boolean conditions associated with the at least two state transitions are evaluated in the specified priority ordering (col 15, lines 20-30).

Per claim 19, Uczekaj et al. disclose:

The method of claim 1, wherein the state diagram information further specifies one or more stop states; wherein said programmatically generating comprises programmatically generating graphical source code such that if during execution of the graphical program one of the stop states becomes active, then the graphical program is caused to stop execution (col 14, lines 40-43; col 14, lines 47-55).

Per claim 20, Uczekaj et al. disclose:

The method of claim 1, further comprising: receiving information specifying a change to the state diagram information; programmatically updating the graphical program to reflect the specified change (col 6, lines 48-67; col 7, lines 1-6; col 10, lines 33-40).

Per claim 21, Uczekaj et al. disclose:

The method of claim 1, wherein said programmatically generating the graphical program comprises calling an application programming interface (API) enabling the programmatic generation of a graphical program (col 16, lines 39-60).

Per claim 22, Uczekaj et al. disclose:

The method of claim 1, wherein said programmatically generating the graphical program comprises programmatically requesting a server program to generate the graphical program (col 3, lines 25-33; col 8, lines 9-23).

Per claims 23, 26, and 29, see the rejection of claim 1 above.

Per claims 24, 27, and 30, see the rejection of claim 7 above.

Per claim 25, Uczekaj et al. disclose:

A computer-implemented method for programmatically generating a graphical program based on a state diagram, comprising: displaying an initial state diagram (col 10, lines 7-22); programmatically generating a graphical program corresponding to the initial state diagram (col 10, lines 23-40); receiving user input specifying a change to the initial state diagram (col 9, lines 46-61; col 7, lines 7-11, "the user has completed entry of the information into object interface section"); programmatically updating the graphical program to correspond to the specified change, in response to the user input specifying the change (col 6, lines 48-67; col 7, lines 1-6; col 10, lines 33-40).

Per claim 28 and 31, see the rejection of claim 8 above.

Per claim 3, Uczekaj et al. disclose the state diagram representing desired operation of a software program ("...creating objects and object control fora drill system ...the invention can be used to describe objects in various other environments", col 4, lines 23-39; col 5, lines 34 -41; col 10, lines 7-22).

Per claim 5, Uczekaj et al. disclose the state diagram for a drill system and that the state diagram can be used to describe objects in various other environments (col 4, lines 23-39; col 5, lines 34 -41; col 10, lines 7-22). A state diagram is used to describe the behavior of a system and each diagram usually represents objects of an individual class and identifies the different states of its objects through the system. As an algorithm is any sequence of operations for performing a specific task, the state diagram can represent the desired algorithm for software, any other non-software system so that each state of operation can be specified, conceptualized, visualized, and constructed in the diagram. Thus, validating and testing the architectural design of the system can be accomplished in a straightforward manner. Therefore, accordingly, Uczekaj et al anticipate this claim.

Per claims 4 and 6, as an algorithm can be designed for anything, see the rejection of claim 5 above.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 11 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Uczekaj et al. (US 5,920,718), further in view of Kodosky et al. (US 5,732,277).

In regards to claim 11, Uczekaj et al. do not specifically disclose that the placeholder graphical source code for each state comprises a case in a graphical case structure. However, Kodosky et al. disclose that the placeholder graphical source code for each state comprises a case in a graphical case structure (col 20, lines 30-49; col 11, lines 43-60, col 11, lines 44-60) so that it is easy for a user to cycle through the alternatives of each case.

Therefore, It would have been obvious to one having ordinary skill in the art at the time of the invention was made to incorporate the teaching of Kodosky et al. to the method of Uczekaj et al. The modification would have been obvious because one having ordinary skill in the art would have been motivated to include a case structure so that a menu list of alternatives on the screen for a user to choose from is available.

In regards to claim 18, Uczekaj et al. do not specifically disclose that the state diagram information specifies an initially active state; wherein said programmatically generating comprises programmatically generating graphical source code such that the graphical program begins execution in the initially active state. However, Kodosky et al. disclose the state diagram information specifying an initial active state and the graphical program beginning execution in the initial active state (col 35, lines 13-18; col 35, lines 31-64) so that the initial graphical program launches displaying the corresponding initial state diagram and the user can simply start to change the state diagram accordingly.

Therefore, It would have been obvious to one having ordinary skill in the art at the time of the invention was made to incorporate the teaching of Kodosky et al. to the method of Uczekaj et al. The modification would have been obvious because one having ordinary skill in the art would have been motivated to set the state initially active so that the initial graphical program corresponding to the initial state diagram is automatically created and the user could simply change the state diagram as desired by adding state or transition, etc.

Response to Arguments

8. Applicant's arguments filed 3/4/2004 have been fully considered but they are not persuasive.

Per claim 1:

The Applicant states that Uczekaj et al. (hereinafter referred to as "Uczekaj") do not disclose generating a graphical program "where a graphical program includes a plurality of interconnected nodes that visually indicate the functionality of the program (page 10, the last paragraph)." In response to applicant's argument that the reference fails to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., a graphical program where a graphical program includes a plurality of interconnected nodes that visually indicate the functionality of the program) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See In re

Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). As such, the claims are read with the broadest reasonable interpretation in mind (Note MPEP 2111).

Uczekaj teaches generating a graphical program generated based on a state diagram ("A display generator retrieves the control information stored in the GUI model and displays it in a predefined location on the display for presenting a state diagram... Once the user is fully satisfied with the information entered into graphical interface tool window... the user activates a generate code button 286 that causes application shell code based on the information entered to be automatically generated. The generated code is called application shell code because all code is generated except the specific code for any user method names entered in define user method section (col 10 lines 23-40; col 3, lines 33-49; col 4, lines 28-39). The examiner interprets a graphical program as a program of or relating to written or visual representation. Therefore, the application shell code generated by activating a generate code button 286 within the graphical interface tool window is interpreted as a graphical program. Accordingly, in view of the broadest reasonable interpretation, Uczekaj discloses programmatically generating a graphical program based on a state diagram as claimed. Therefore, the rejection of claim 1 is considered proper and maintained.

Per claims 23, 25, 26 and 29:

The applicant states that Uczekaj does not disclose the limitations of claims 23, 25, 26 and 29, for the reasons set forth in connection with claim 1. As shown above, the

Art Unit: 2124

rejection of claim 1 by Uczekaj was maintained, and accordingly, the rejections of claims 23, 25, 26 and 29 are also maintained.

Per claims 2-10, 12-17 and 19-22, 24, 27, 28, 30 and 31:

The applicant states that claims 2-10, 12-17 and 19-22, 24, 27, 28, 30 and 31 are allowable as being dependent on allowable base claims. As has been shown above, the rejections of the independent claims 1, 23, 26 and 29 by Uczekaj are proper, the argument that claims 2-10, 12-17 and 19-22, 24, 27, 28, 30 and 31 are allowable as being dependent on an allowable base claim is considered moot.

Accordingly, the rejections of claims 2-10, 12-17 and 19-22, 24, 27, 28, 30 and 31 are proper and maintained.

Per claim 11:

Claim 11 further recites a case structure.

The applicant argued claim 11 is non-obvious over Uczekaj in view of Kodosky because "Nor is there any teaching or suggestion in Uczekaj to include the limitation that the placeholder graphical source code for each state comprises a case in a graphical case structure" and "Nor is there any teaching or suggestion in Kodosky to include the limitation of "programmatically generating the graphical program in response to the state diagram information, wherein said programmatically generating comprises programmatically generating graphical source code corresponding to the specified one or more states." It is noted that Kodosky uses a case structure, which is well known

Art Unit: 2124

programmatic structure in the art of software development (see "Case (Conditional) Selection Structure," col 20, lines 30-49; col 11, lines 43-60, col 11, lines 44-60).

Furthermore, the applicant fails to show that the reasons to combine and motivations concerning the rejection of claim 11 are improper. As has been shown above, the rejections of independent claims 1, 23, 25, 26 and 29 by Uczekaj are proper, and Kodosky discloses the case structure. As such, in view of the combined teachings by Uczekaj and Kodosky, the rejection of claim 11 is proper and maintained.

Per claim 18:

The applicant argued, "Kodosky's state information is execution state information for the graphical program nodes themselves, and so is not usable to generate the graphical program." It is noted that claim 18 recites a limitation: state diagram specifying an initially active state and execution beginning in the initially active state, which is well known in the art of software development and is disclosed by Kodosky. The applicant fails to show that the reasons to combine and motivations concerning the rejections of claim 18 are improper. The applicant's argument is based on a single reference, Kodosky, not in combination with teachings by Uczekaj. As has been shown above, the rejections of independent claims 1, 23, 25, 26 and 29 by Uczekaj are proper and Kodosky discloses state diagram specifying an initially active state and execution beginning in the initially active state ("when a diagram begins execution it set to its active state," col 35 lines 12-20). Accordingly, in view of the combined teachings of Kodosky and Uczekaj, the rejection of claim 18 is proper and maintained.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Insun Kang whose telephone number is 703-305-6465. The examiner can normally be reached on M-F 8:30-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on 703-305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2124

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

IK
5/12/2003


KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100